**End of Result Set**

□ ▓▓ Generate Collection ▓▓ | Print |

L10: Entry 5 of 5                      File: USPT                   Jun 2, 1998

DOCUMENT-IDENTIFIER: US 5761654 A
**\*\* See image for Certificate of Correction \*\***
TITLE: Memory structure and method for tuning a database statement using a join-tree data structure representation, including selectivity factors, of a master table and detail table

Abstract Text (1):
A method of constructing an aid to tuning of database statements comprises a data structure (stored in a memory on a computer system) which compactly represents that information which is needed about a database statement to determine the optimal series of operations (e.g., table data fetches) needed to execute the statement. The information includes the relationships between tables joined in the statement and the selectivities of logical conditions applied to rows in those tables.

Brief Summary Text (19):
Typically a user, either human or machine, sends an SQL statement to a database management program with SQL search capabilities, such as Oracle7 (.TM.), distributed by the assignee of this application, and other programs well known to those of ordinary skill. Such a program is referred to for convenience as a "search program." The search program interprets the SQL statement and combs through the database in search of information satisfying the statement. In the hypothetical database above, for example, a user might want to query the database to create a list of customers serviced by a particular sales representative.

Detailed Description Text (14):
select sum(decode(sign(sal-100000), 1, 1, 0))/count(\*) from emp or using similar statements in other SQL dialects.

Current US Original Classification (1):
707/2

Current US Cross Reference Classification (1):
707/101

Current US Cross Reference Classification (2):
707/3

Current US Cross Reference Classification (3):
707/4

First Hit    Fwd Refs

☐ ▓ Generate Collection ▓  Print


L10: Entry 3 of 5                    File: USPT                    Aug 22, 2000


DOCUMENT-IDENTIFIER: US 6108664 A
TITLE: Object views for relational data


Detailed Description Text (44):
At step 410, database server 202 receives a request to define an object view that
specifies a key-based OID. For example, database server 202 an SQL statement
similar in form to the OV.sub.-- EMPLOYEE definition represented by FIG. 3E.

Detailed Description Text (50):
At step 430, database server 210 receives a request for objects based on an object
view on a relational table. The object view specified by the request is referred to
as the requested object view. Assume, for example, that user 208 transmits the
following SQL statement to database server 202.

Detailed Description Text (82):
A statement issued for the purpose of defining a trigger is referred to as a
trigger creation statement. In response to receiving the above trigger creation
statement, database server 202 defines the trigger by adding to the data definition
a trigger definition as specified by the SQL statement. In this example, the SQL
statement specifies that a trigger named TR.sub.-- AFTER.sub.-- EMPLOYEE is to be
invoked after the database server 202 performs a direct update on relational table
FORMER.sub.-- EMPLOYEE.

Detailed Description Text (92):
In response to receiving the above trigger creation statement, database server 202
defines the trigger by adding to the data definition a trigger definition as
specified by the trigger creation statement. By adding a trigger definition to the
data definition of the view, the database server 202 is associating the trigger
with the view. The trigger definition includes data indicating the trigger event
associated with the instead-of trigger. The computer instructions associated with
the trigger, or a reference to such instructions, is also included in trigger
definition. A database server user 202, converts the block(s) of code contained in
a trigger creation statement into a form of intermediate instructions that may be
later executed by the database server 202 when the trigger is invoked.

Current US Original Classification (1):
707/103R

Current US Cross Reference Classification (1):
707/102

Current US Cross Reference Classification (2):
707/104.1

☐ | Generate Collection | | Print |

L10: Entry 2 of 5                          File: USPT                          Sep 24, 2002


DOCUMENT-IDENTIFIER: US 6457007 B1
TITLE: Distributed database management system including logical database
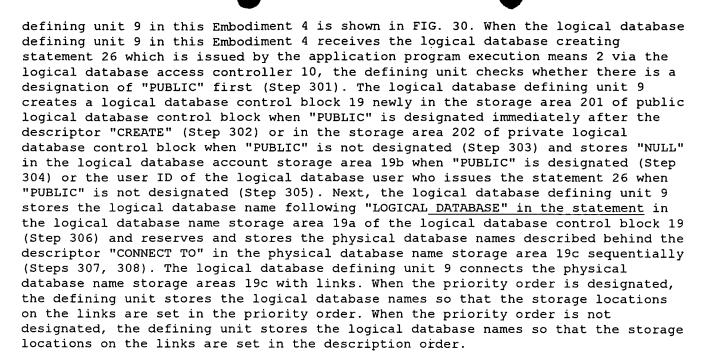constituted by a group of physical databases


Brief Summary Text (10):
As a method for realizing such a subject, there is a derivation database processing
method which is described in Japanese Patent Application Laid-Open No. 4-112246.
This method collects tables on a plurality of distributed database management
systems partially and handles them as a derivation database and the user registers
the definition information of each database management system and the definition
information of the derivation database in a dictionary. The definition information
of each database management system consists of identification information of a
plurality of tables stored in each database and the definition information of the
derivation database consists of identification information of a plurality of
database management systems and tables which constitute the derivation database.
The database manipulation means receives a derivation database access request from
the application program, searches the above dictionary for the table location, and
issues a database access statement to the database management system where the
table is stored. By this method, the application program can access a plurality of
database management systems via a derivation database.

Detailed Description Text (51):
When the logical database defining unit 9 receives the logical database creating
statement 14 issued from the application program execution means 2 via the logical
database access controller 10, the unit checks first whether there is a designation
of PUBLIC (Step 311). When PUBLIC is designated immediately after the descriptor
CREATE, the logical database defining unit 9 creates a logical database control
block 13 newly in the storage area 201 of public logical database control block
(Step 312). When PUBLIC is not designated, the logical database defining unit 9
creates a logical database control block 13 newly in the storage area 202 of
private logical database control block (Step 313). Next, the logical database
defining unit 9 stores the logical database name following LOGICAL DATABASE in the
statement in the logical database name storage area 13a of the logical database
control block 13 (Step 314) and stores the logical database user name following
USER_ID and the logical database password following IDENTIFY sequentially (Step
310) in the logical database account storage area 13c (Step 315). Furthermore, the
logical database defining unit 9 stores the physical database name described behind
DATABASE sequentially (Step. 319) in the physical database name storage area 13d
(Step 316) and stores the user ID of the USER-ID clause following the above
physical database name and the password of the IDENTIFY clause in the physical
database account storage area 13e corresponding to the above physical database name
storage area 13d (Step 317). The logical database defining unit 9 connects the
physical database name storage areas 13c with links and stores the above data so
that the storage locations on the links are set in the description order. Finally,
the logical database defining unit 9 stores the ID of the OS user who issues the
statement and the use or change permission kind in the access privilege storage
area 13b (Step 318).

Detailed Description Text (148):
The flow of the logical database registration processing of the logical database

defining unit 9 in this Embodiment 4 is shown in FIG. 30. When the logical database defining unit 9 in this Embodiment 4 receives the logical database creating statement 26 which is issued by the application program execution means 2 via the logical database access controller 10, the defining unit checks whether there is a designation of "PUBLIC" first (Step 301). The logical database defining unit 9 creates a logical database control block 19 newly in the storage area 201 of public logical database control block when "PUBLIC" is designated immediately after the descriptor "CREATE" (Step 302) or in the storage area 202 of private logical database control block when "PUBLIC" is not designated (Step 303) and stores "NULL" in the logical database account storage area 19b when "PUBLIC" is designated (Step 304) or the user ID of the logical database user who issues the statement 26 when "PUBLIC" is not designated (Step 305). Next, the logical database defining unit 9 stores the logical database name following "LOGICAL DATABASE" in the statement in the logical database name storage area 19a of the logical database control block 19 (Step 306) and reserves and stores the physical database names described behind the descriptor "CONNECT TO" in the physical database name storage area 19c sequentially (Steps 307, 308). The logical database defining unit 9 connects the physical database name storage areas 19c with links. When the priority order is designated, the defining unit stores the logical database names so that the storage locations on the links are set in the priority order. When the priority order is not designated, the defining unit stores the logical database names so that the storage locations on the links are set in the description order.
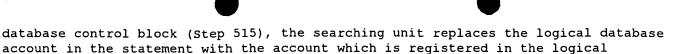
Detailed Description Text (151):
The flow of the logical database changing processing of the logical database defining unit 9 in this Embodiment 4 is shown in FIG. 32. When the logical database defining unit 9 in this Embodiment 4 receives the logical database changing statement 27 which is issued by the application program execution means 2 via the logical database access controller 10, the defining unit checks whether there is a designation of "PUBLIC" first and searches the logical database control block 19, which holds the logical database name following "LOGICAL DATABASE" in the statement in the logical database name storage area 19a, in the storage area 201 of public logical database control block when designated or in the storage area 202 of private logical database control block when not designated (Step 320).

Detailed Description Text (171):
Furthermore, the logical database defining unit 9 checks whether there is a description "ADD" in the statement 31. When there is "ADD", the defining unit reserves the physical database name storage area 20c and the physical database account storage area 20d corresponding to the above storage area 20c in the logical account control block 20 which is detected at Step 330 sequentially for all the databases described in the statement 31, stores the physical database name which is described immediately after "ADD" in the physical database name storage area 20c in which an area is newly reserved, and furthermore stores the physical, database user ID and physical database password which are described behind the physical database name in the physical database account storage area 20d in which an area is newly reserved (Step 332). The logical database defining unit 9 connects the physical database name storage area 20c and physical database account storage area 20d which are newly reserved and the existing storage areas 20c and 20d with links by linking them so that the newly reserved storage areas are positioned on the last link.

Detailed Description Text (197):
Furthermore, in Embodiment 4 or 5, even when a user accesses the physical database 1 which is designated directly by him by using an access statement in which the physical database 1 is directly designated, he can issue a CONNECT statement using the logical database account. By doing this, the user can directly designate and access one of a plurality of physical databases 1 by using an account without being aware of differences in account between the physical databases 1. The reason is that when the table location searching unit 7 in Embodiment 4 or 5 transfers a CONNECT statement having a database name which is not registered in the logical

database control block (Step 515), the searching unit replaces the logical database account in the statement with the account which is registered in the logical account control block 20. It will be explained with reference to FIGS. 10 and 17.

Detailed Description Text (306):
As mentioned above, according to the logical database access function in this Embodiment 11, by a SELECT statement which is similar to a join for tables stored in a single physical database 1, a plurality of tables stored in different physical databases 1 in the same logical database can be joined.

Current US Original Classification (1):
707/10

Current US Cross Reference Classification (1):
707/100

CLAIMS:

13. A distributed database management system which is an information processor which is connected to at least one physical database holding a table via a physical database management system for managing said physical database, and is connected furthermore to application program execution means for accessing said physical database, receiving an access statement for accessing physical databases which is issued by said application program execution means, selecting one of said physical databases as an access object, and transferring said access statement to the physical database management system for managing said physical database, comprising: a logical database dictionary for holding information on a logical database in which at least one of said physical databases is grouped; a logical database defining unit for registering information on said logical database in said logical database dictionary; a statement buffer for holding said access statement; a logical database access controller for receiving an access statement from said application program execution means and storing it in said statement buffer; and a table location searching unit for transferring said access statement held in said statement buffer to the physical database management system for managing said physical database by selecting one of said physical databases as an access object; wherein said table location searching unit includes means for receiving an access statement for instructing logging in the database, for selecting one of said physical databases belonging to the instructed logical database as an access object of said access statement when the database which is instructed to be connected is registered in said logical database dictionary as said logical database, for selecting another one of said physical databases belonging to the instructed logical database as an access object of said access statement when a response of access refusal is given to the transfer of said access statement, and for transferring said access statement held in said statement buffer once again.

41. A distributed database management method in a system which is an information processor which is connected to at least one physical database holding a table via a physical database management system for managing said physical database, and is connected furthermore to application program execution means for accessing said physical database, receiving an access statement for accessing physical databases which is issued by said application program execution means, selecting one of said physical databases as an access object, and transferring said access statement to the physical database management system for managing said physical database, said method comprising the steps of: holding in a logical database dictionary, information on the logical database in which at least one of said physical databases is grouped; registering by a logical database defining unit, information on said logical database in said logical database dictionary; holding in a statement buffer, said access statement; receiving through a logical database access controller, an access statement from said application program execution means and storing it in said statement buffer; and transferring through a table

location searching unit, said access statement held in said statement buffer to the physical database management system for managing said physical database by selecting one of said physical databases as an access object; wherein said table location searching unit includes means for receiving an access statement for instructing logging in the database, for selecting one of said physical databases belonging to the instructed logical database as an access object of said access statement when the database which is instructed to be connected is registered in said logical database dictionary as said logical database, for selecting another one of said physical databases belonging to the instructed logical database as an access object of said access statement when a response of access refusal is given to the transfer of said access statement, and for transferring said access statement held in said statement buffer once again.